Chapter 11

Implementing Nonlinear Approximations Empirically

THE PREVIOUS CHAPTER PRESENTED methodologies for obtaining nonlinear model approximations. In this chapter, we illustrate how the full range of empirical methodologies presented in part II of the text can be used to analyze the model under investigation once a nonlinear approximation has been obtained. Applications of calibration and moment-matching exercises can be based on model simulations, and classical and Bayesian full-information analyses can be pursued by substituting a particle filter for the Kalman filter as a means of evaluating the likelihood function.

To briefly reestablish notation, recall that s_t denotes the vector of state variables of the model. Its law of motion is given by

$$s_t = f(s_{t-1}, \upsilon_t), \tag{11.1}$$

where υ_t represents the collection of structural shocks incorporated in the model. Further, c_t denotes the vector of control variables of the model. The policy function for c_t is given by

$$c_t = c(s_t); \tag{11.2}$$

its approximated counterpart is given by

$$c_t = \widehat{c}(s_t). \tag{11.3}$$

The full collection of model variables are contained in the vector $x_t = (s_t', c_t')'$. Finally, for empirical implementation, (11.1) and (11.3) are mapped into observables by the observation equation

$$\begin{aligned} X_t &= \widetilde{g}(s_t, c_t, \upsilon_t, u_t) \\ &\equiv g(s_t, u_t), \end{aligned} \tag{11.4}$$

where u_t represents measurement error.

11.1 Model Simulation

The recursive nature of DSGE models renders the task of model simulation straightforward (for details regarding the accuracy of simulations involving

DSGE models, see Santos and Peralta-Alva, 2005). The list of inputs to the simulation process includes a specification of the initial state vector s_0, which can be taken either as known or as a realization obtained from an underlying distribution. The stationary nature of the model (typically induced by an appropriate transformation of variables) renders this distinction as unimportant in simulations designed to characterize the asymptotic properties of the model. Additional inputs include the law of motion (11.1) for s_t, the approximated policy function $\widehat{c}(s_t)$, the mapping (11.4) from model variables x_t to observables X_t, and a random number generator used to obtain artificial drawings of stochastic shocks (and if appropriate, measurement errors) from their underlying distributions. The parameterization of these inputs is determined by a specific specification chosen for the structural parameters of the underlying model, denoted by μ.

Given these inputs, model simulation proceeds exactly as in the linear case. From (11.1), s_0 and $\{\upsilon_t\}_{t=1}^T$ yield $\{s_t\}_{t=1}^T$ directly. Using this series as an input to the policy function yields $\{c_t\}_{t=1}^T$. Then $\{s_t\}_{t=1}^T$ and $\{c_t\}_{t=1}^T$ combine to form $\{x_t\}_{t=1}^T$ ($x_t \equiv (s_t', c_t')$), which combined with $\{u_t\}_{t=1}^T$ maps into X_t as in (11.4). Functions of interest can then be computed using the simulated realizations of $\{X_t\}_{t=1}^T$. To eliminate the influence of the starting value chosen for s_0, a burn-in phase can be implemented. This merely involves discarding the first, say 1,000 artificial realizations obtained for $\{s_t\}_{t=1}^T$. Finally, numerical standard errors can be virtually eliminated using a sufficiently large number of artificial realizations (see chapter 9 for details regarding numerical standard errors).

A novelty in this case arises when using a Markov process to represent the evolution of a stochastic process. For example, recall from the specification of the optimal growth model used in chapter 10 that a Markov process was used to represent the behavior of total factor productivity (TFP) $\{a_t\}_{t=1}^T$. A simple algorithm for generating artificial realizations of $\{a_t\}_{t=1}^T$ for the two-state case is as follows (extension to the n-state case is straightforward). Begin with a particular specification of ξ_0, the 2×1 vector with a 1 in the first row and a zero in the second row given state one (e.g., the low state for a_t), and a 0 in the first row and a 1 in the second row given state two (e.g., the high state). Set a_0 accordingly (e.g., either $a_0 = aL$ or $a_0 = aH$). With P again denoting the state-transition matrix, the probabilities of realizing states one and two next period are given by $P\xi_0 = [p_1, p_2]'$. To determine the outcome of this random event, draw a random variable u distributed uniformly over the $[0, 1]$ interval (e.g., using the GAUSS command `rndu`), and infer the realization of state one if $p_1 > u$ and state two otherwise. Update ξ_1 accordingly and repeat until T realizations of a_t are obtained.

To calibrate the Markov process to match an AR(1) specification

$$a_t = (1 - \rho)\overline{a} + \rho a_{t-1} + \varepsilon_t, \qquad s.e.(\varepsilon) = \sigma_\varepsilon, \tag{11.5}$$

TABLE 11.1
Model Simulations

	Linear Approximation				*Nonlinear Approximation*			
j	σ_j	$\frac{\sigma_j}{\sigma_y}$	$\varphi(1)$	$\varphi_{j,y}(0)$	σ_j	$\frac{\sigma_j}{\sigma_y}$	$\varphi(1)$	$\varphi_{j,y}(0)$
y	0.0199	1.00	0.94	1.00	0.0151	1.00	0.89	1.00
c	0.0156	0.78	0.96	0.99	0.0109	0.72	0.96	0.95
i	0.0350	1.76	0.90	0.99	0.0330	2.19	0.79	0.94
k	0.0348	1.75	0.98	0.95	0.0184	1.22	0.99	0.76
a	0.0112	0.56	0.80	0.88	0.0112	0.74	0.80	0.94

Note: $\varphi(1)$ denotes first-order serial correlation; $\varphi_{j,y}(0)$ denotes contemporaneous correlation between variables j and y. Model moments based on the parameterization

$$\mu = [\alpha\ \beta\ \delta\ \phi\ \rho\ \sigma]' = [0.33\ 0.96\ 0.1\ 2.0\ 0.8\ 0.0067]'.$$

set the diagonal elements of P as

$$p_{11} = p_{22} = \frac{\rho + 1}{2}, \tag{11.6}$$

and the values for a_t in states one and two as

$$a_1 = \overline{a} - \sigma_a, \qquad a_2 = \overline{a} + \sigma_a, \tag{11.7}$$

$$\sigma_a = \sigma_e \sqrt{\left(\frac{1}{1 - \rho^2}\right)}. \tag{11.8}$$

Under this specification, a_t bounces one standard deviation above and below its mean value $\overline{a}$.

11.1.1 Simulating the Optimal Growth Model

To demonstrate differences in model inferences that can arise in working with nonlinear rather than linear model approximations, table 11.1 presents a collection of moments calculated using linear and nonlinear approximations of the stochastic growth model presented in chapter 10. Both sets of simulations are based on the use of steady state values for s_0, 1,000 burn-in drawings, and 10,000 retained drawings. Simulations of the linear approximation were based on the use of an AR(1) specification for a_t, under the assumption of normality for its innovations υ_t. Simulations of the nonlinear approximation were based on the use of a two-state Markov process, as described above. In both cases, moments were calculated for logged deviations of the model variables from steady state values.

Although many of the moment calculations are similar across solution methods, there are some notable differences. Most obviously, k is much

less volatile relative to y under the nonlinear approximation, and is also less-strongly correlated with y. The opposite pattern is true of the relationship between a and y. This seems to have more to do with the linear approximation of the production function rather than the policy function, because differences in the relationship between c, i, and y across solution methods are less distinct. One difference across simulation methodologies that does not account for these differences in moment calculations is the alternate AR and Markov specifications employed for a_t used under the linear and non-linear approximations, as the following exercise demonstrates.

Exercise 11.1

As an alternative to the two-state Markov process specified for a_t under the nonlinear approximation, reconstruct table 11.1 using an AR(1) specification, parameterized as indicated in table 11.1, along with the associated policy function approximation constructed using `ara.prg`. Under this exercise, you should be able to match closely the calculations reported in the table.

11.2 Full-Information Analysis Using the Particle Filter

We now describe how full-information analyses can be conducted using nonlinear model approximations by substituting a particle filter for the Kalman filter as a means of evaluating the likelihood function. The application of particle filters to the analysis of DSGE models has been advanced by Fernandez-Villaverde and Rubio-Ramirez (2004b, 2005), and a textbook reference on particle filters is available from Doucet, deFreita, and Gordon (2001). Just as the case with nonlinear solution methods, many alternative specific algorithms can be used to implement particle filters. Pitt and Shephard (1999) provide an overview of alternative algorithms, and demonstrate applications to ARCH and stochastic-volatility models. Here we follow Fernandez-Villaverde and Rubio-Ramirez in focusing on a specific algorithm with close ties to the Monte Carlo integration technique of importance sampling, described in detail in chapter 9.

11.2.1 Overview

As we have seen in chapters 8 and 9, full-information analyses entail calculations of the probability or likelihood associated with the realization of an observed sample $X \equiv \{X_t\}_{t=1}^{T}$. In chapter 4, section 4.3, we characterized the Kalman filter as an algorithm designed to execute this calculation

recursively, following the recursive nature of its associated structural model. Although the specific implementation of the Kalman filter is specialized to the case in which the underlying state-space representation is linear and structural innovations and measurement errors are normally distributed, the underlying algorithm can be applied more generally. Indeed, the particle filter retains this algorithm; only the details regarding implementation differ.

Recall that the idea behind the algorithm is to produce assessments of the conditional probability associated with the time-t observation X_t, given the history of past realizations $X^{t-1} \equiv \{X_j\}_{j=1}^{t-1}$. Denote this probability as $L(X_t|X^{t-1})$, with $L(X_1|X^0)$ denoting the unconditional likelihood associated with X_1. The sequence of conditional likelihoods $\{L(X_t|X^{t-1})\}_{t=1}^{T}$ are independent across time, thus the likelihood associated with X is given by the product of the individual conditional likelihoods:

$$L(X) = \prod_{t=1}^{T} L(X_t|X^{t-1}).$$

Regarding the structure of $L(X_t|X^{t-1})$, this is most simply described for the case in which each of the elements of x_t, including s_t, is observable. Conditional on $\{s_j\}_{j=1}^{t-1}$, from (11.1) we observe that the optimal forecast of s_t is given by

$$\widehat{s}_t = f(s_{t-1}, 0).$$

Moreover, the inferred realization of υ_t is that which reconciles the difference between the forecasted and observed value of s_t; that is, $\widehat{\upsilon}_t$ is constructed to satisfy

$$\widehat{s}_t - f(s_{t-1}, \widehat{\upsilon}_t) = 0.$$

The conditional likelihood associated with the observation of X_t can thus be assessed as the likelihood assigned to $\widehat{\upsilon}_t$ by its assumed probability distribution (say, p_υ):

$$L(X_t|X^{t-1}) = p_\upsilon(\widehat{\upsilon}_t).$$

As with the Kalman filter, the details behind the particle filter are slightly more complicated when certain elements of x_t are unobservable, but the basic idea is the same: conditional likelihoods represent probabilities associated with the realization of observables at time t, given the sequence of variables that were observed previously.

11.2.2 Case 1: No Measurement Error

We begin with the case in which no measurement error is associated with the observation of X_t. In this case the model can be expressed as

$$s_t = f(s_{t-1}, \upsilon_t) \tag{11.9}$$

$$X_t = g(s_t), \tag{11.10}$$

where the policy function $c(s_t)$ is subsumed in (11.9) and/or (11.10). The specification of the model is closed with a distributional assumption for the structural shocks υ_t, the dimension of which matches the dimension of X_t to avoid a structural singularity problem. Let this distribution be given by $p(\upsilon_t)$. Once again, the parameterizations of $f(s_{t-1}, \upsilon_t)$, $g(s_t)$, and $p(\upsilon_t)$ are determined by the specification of the parameters μ.

LIKELIHOOD CONSTRUCTION

Full-information analysis regarding the specification of μ is accomplished through the analysis of the likelihood function $L(X^T|\mu)$, where $X^T = \{X_j\}_{j=1}^T$. An overview of the construction of $L(X^T|\mu)$ is as follows. First, (11.9) and (11.10) are used to obtain an expression for υ_t as a function of $X^t = \{X_j\}_{j=1}^t$, s_0, and μ. This expression enables inference regarding the behavior of the unobservable structural shocks, conditional on the parameterized model, the observed data, and s_0. The probability assigned to this behavior by $p(\upsilon_t)$ provides the basis upon which the probability associated with the corresponding specification of μ is assessed.

To derive the desired expression for υ_t, begin by solving for s_t in (11.10):

$$s_t = g^{-1}(X_t). \tag{11.11}$$

Next, substitute for s_t in (11.9) to obtain

$$g^{-1}(X_t) = f(s_{t-1}, \upsilon_t). \tag{11.12}$$

Finally, solve for υ_t to obtain

$$\upsilon_t = \upsilon(X_t, s_{t-1}). \tag{11.13}$$

At this point, (11.13) can be combined with the updating equation (11.9) to establish a recursion for constructing the sequences $s^T = \{s_t\}_{t=1}^T$ and $\upsilon^T = \{\upsilon_t\}_{t=1}^T$ as functions of X^T and s_0. This proceeds as follows. Along with a given s_0, insert X_1 into (11.13) to obtain

$$\upsilon_1 = \upsilon(X_1, s_0). \tag{11.14}$$

Next, insert the inferred value of υ_1 into (11.9) to calculate the transition it implies from s_0 to s_1:

$$s_1 = f(s_0, \upsilon(X_1, s_0)). \tag{11.15}$$

For the subsequent steps $t = 2, \ldots, T$, υ_t is constructed using X_t and s_{t-1} as inputs in (11.13), and s_t is updated using s_{t-1} and υ_t as inputs in (11.9). Hereafter, we will denote the resulting sequences as $s^T(X^T, s_0)$ and $\upsilon^T(X^T, s_0)$, with t^{th} elements denoted as $s_t(X^t, s_0)$ and $\upsilon_t(X^t, s_0)$.

Because υ_t is serially uncorrelated by assumption, the likelihood function for X^T, which at this point is conditional upon s_0, is constructed as the product of the individual likelihoods $p(\upsilon_t(X^t, s_0))$:

$$L(X^T|s_0, \mu) = \prod_{t=1}^{T} p(\upsilon_t(X^t, s_0)). \tag{11.16}$$

To eliminate conditionality on s_0, $L(X^T|s_0, \mu)$ is integrated over the distribution for s_0 implied by the specification of the model and the observed data. Denoting this distribution as $p(s_0|X^t)$, the unconditional likelihood function is given by

$$L(X^T|\mu) = \prod_{t=1}^{T} \int p(\upsilon_t(X^t, s_0)) p(s_0|X^t) ds_0. \tag{11.17}$$

Consider the construction of the likelihood function associated with the optimal growth model. Recall that in this case the state is $s_t = [a_t \;\; k_t]'$, and the single structural shock is $\upsilon_t = \varepsilon_t$, where ε_t is the innovation to TFP appearing in (10.8). The distribution of this innovation is given by $p(\varepsilon_t)$. Let output y_t serve as the single observable variable in this example, and take the availability of the policy function for consumption, denoted as

$$c_t = c(a_t, k_t), \tag{11.18}$$

as having been determined via an approximation scheme. In this case, (11.9) is given by the transition equations

$$\ln(a_t) = \rho \ln(a_{t-1}) + \varepsilon_t, \tag{11.19}$$

$$k_t = a_{t-1} k_{t-1}^{\alpha} - c(a_{t-1}, k_{t-1}) + (1 - \delta) k_{t-1}. \tag{11.20}$$

Also, the observation equation (11.10) is given by

$$y_t = a_t k_t^{\alpha}. \tag{11.21}$$

Substituting for a_t and k_t in (11.21) using (11.19) and (11.20), respectively, and then solving for ε_t, we obtain

$$\varepsilon_t = \ln\left[\frac{y_t}{e^{(\rho a_{t-1})}(a_{t-1}k_{t-1}^{\alpha} - c(a_{t-1}, k_{t-1}) + (1-\delta)k_{t-1})^{\alpha}}\right]$$
$$\equiv \varepsilon(y_t, a_{t-1}, k_{t-1}). \tag{11.22}$$

Given $s_0 = [a_0 \; k_0]'$ and y_1, the implied value of ε_1 can be obtained using (11.22). Next, the implied values of $s_1 = [a_1 \; k_1]'$ can be obtained by inserting $(a_0 \; k_0 \; \varepsilon_1)$ into the transition equations (11.19) and (11.20). Repeating this process for $t = 2, \ldots, T$ yields $\{\varepsilon_t\}_{t=1}^{T}$, and thus the sequence $\{p(\varepsilon_t)\}_{t=1}^{T}$.

LIKELIHOOD EVALUATION

Returning to the general model representation, inferences made conditionally upon s_0 do not require the calculation of an integral in the likelihood evaluation step. All that is required in this case is the construction of $\upsilon^T(X^T, s_0)$ as described above, which immediately enables the evaluation of $L(X^T|s_0, \mu)$ in (11.16). However, if the value of s_0 is unknown, the integral in (11.17) must be calculated, and the particle filter comes into play.

Before describing the particle filter, we pause to discuss the conditionality of the distribution $p(s_0|X^t)$ on X^t. This conditionality is important: introduction of the particle filter is unnecessary in its absence (for reasons we will explain shortly). The intuition behind conditionality is perhaps best seen by reverting to the example of the optimal growth model, and considering (11.22). Suppose y_1 is large relative to its steady state value. Then clearly, alternative candidate values of $[a_0 \; k_0]'$ are not equally likely. In particular, the specification of relatively small values for $[a_0 \; k_0]'$ imply the realization of very large TFP innovations ε_1, which are assigned relatively little weight from their corresponding distribution $p(\varepsilon_t)$. Moreover, because the model in question will typically embody a nontrivial degree of persistence, the influence of $[a_0 \; k_0]'$ on inferences regarding ε_t will extend beyond the first period. Thus the plausibility of a particular specification of $[a_0 \; k_0]'$ will tend to hinge not only upon y_1, but in general on y^t.

Returning again to the general model representation, suppose for illustrative purposes that $p(s_0|X^t)$ is not conditional upon X^t. In this case, a simple numerical integration algorithm can be used to approximate the integral appearing in (11.17). A single step in the algorithm involves obtaining a drawing s_0 from its unconditional distribution $p(s_0)$, coupling this drawing with X^T to construct $\upsilon^T(X^T, s_0)$, and then calculating the likelihood value

$$L(X^T|s_0,\mu) = \prod_{t=1}^{T} p(\upsilon_t(X^t, s_0)).$$

Let the likelihood value associated with the i^{th} of N such drawings be given by $L(X^T|s_0^i,\mu)$. Then the approximation to (11.17) we seek is given by the sample average

$$\overline{L(X^T,\mu)}_N = \frac{1}{N}\sum_{i=1}^{N} L(X^T|s_0^i,\mu). \tag{11.23}$$

This algorithm is directly related to that used to approximate the posterior mean of some function $g(\mu)$ for the case in which it is possible to obtain drawings of μ directly from its associated posterior distribution $P(\mu|X^T)$. See chapter 9, section 9.2, for details regarding this algorithm.

As discussed in chapter 9, one means of overcoming the inability to obtain drawings of μ directly from $P(\mu|X^T)$ is the use of an importance sampling algorithm. Briefly, this involves obtaining drawings of μ from a stand-in distribution $I(\mu)$, and approximating $Eg(\mu)$ using the weighted average

$$\overline{g(\mu)}_N = \frac{\sum_{i=1}^{N} g(\mu^i) w(\mu^i)}{\sum_{i=1}^{N} w(\mu^i)}, \tag{11.24}$$

where the weighting function is given by

$$w(\mu) = \frac{P(\mu|X^T)}{I(\mu)}. \tag{11.25}$$

The purpose of the weights is to offset the influence of $I(\mu)$ on the resulting inferences; given their use, the sample drawings $\{\mu^i\}_{i=1}^N$ can be thought of as having been obtained from the posterior distribution itself. See chapter 9, section 9.4.1, for details.

THE PARTICLE FILTER

Recall that our objective is to evaluate the likelihood function $L(X^T|\mu)$ in (11.17), which requires approximations of the sequence of integrals

$$\int p(\upsilon_t(X^t, s_0)) p(s_0|X^t) ds_0, \quad t = 1, \ldots, T.$$

The particle filter we now describe can be thought of as a particular type of importance sampling algorithm designed to approximate these integrals.

Its use in place of (11.23) becomes necessary given the conditionality of $p(s_0|X^t)$ on X^t, which generally eliminates the ability to obtain drawings of s_0 directly from $p(s_0|X^t)$.

To describe the particle filter, we require additional notation. Let $s_0^{t,i}$ denote the i^{th} of N drawings of s_0 obtained from the conditional distribution $p(s_0|X^t)$. A single drawing $s_0^{t,i}$ is referred to as a particle, and the sequence $\{s_0^{t,i}\}_{i=1}^N$ is referred to as a swarm of particles. Also, let $\{s_0^{0,i}\}_{i=1}^N$ be a particle swarm obtained from the unconditional distribution of s_0, denoted as $p(s_0|X^0)$, with X^0 indicating an absence of observations on X. We note that if $p(s_0|X^0)$ is unknown, it may be approximated, for example, using a uniform distribution centered on the steady state value s^*.

Consider first the integral for $t=1$. Using the unconditional density $p(s_0|X^0)$, begin by generating a particle swarm $\{s_0^{0,i}\}_{i=1}^N$. Combining each particle $s_0^{0,i}$ with $X_1 \equiv X^1$ in (11.14), obtain the associated sequence

$$\left\{\upsilon_1\left(X^1, s_0^{0,i}\right)\right\}_{i=1}^N.$$

Note that the likelihood value associated with $s_0^{0,i}$ is given by $p(\upsilon_1(X^1, s_0^{0,i}))$. The average value of these associated likelihood values yields the approximation we seek for $t=1$:

$$\int p(\upsilon_1(X^1, s_0))p(s_0|X^1)ds_0 \approx \frac{1}{N}\sum_{i=1}^N p(\upsilon_1(X^1, s_0^{0,i})). \tag{11.26}$$

Although we have now approximated the integral we seek for $t=1$, an additional step is required before moving to $t=2$. This involves the construction of an approximation of the conditional density $p(s_0|X^1)$. To construct this density, let q_1^i denote the *relative* likelihood value associated with $s_0^{0,i}$, where relativity is with respect to its counterparts in the swarm:

$$q_1^i = \frac{p(\upsilon_1(X^1, s_0^{0,i}))}{\sum_{j=1}^N p(\upsilon_1(X^1, s_0^{0,j}))}. \tag{11.27}$$

In the particle filter, the role of q_1^i is analogous to that of the weighting function in an importance sampling algorithm.

Now, let $\{s_0^{1,i}\}_{i=1}^N$ denote a second sequence of drawings from the original swarm $\{s_0^{0,i}\}_{i=1}^N$. This second swarm is obtained by drawing with replacement from the original swarm, with the probability of obtaining a drawing of $s_0^{0,i}$ determined by q_1^i. Under mild regularity conditions (e.g., as described by Fernandez-Villaverde and Rubio-Ramirez, 2004b), this

second swarm of particles represents a drawing from the conditional distribution $p(s_0|X^1)$.

For each particle in this second swarm, along with its associated structural shock $\upsilon_1(X^1, s_0^{1,i})$, obtain the corresponding value of $s_1^{1,i}$ using the updating equation (11.9). Combine this with $s_0^{1,i}$ to establish the sequence

$$s^1(X^1, s_0^{1,i}) = \left[s_1^{1,i}\ s_0^{1,i}\right]'.$$

This will ultimately contain $T+1$ components, representing the time series of state variables implied by X^T and s_0. The swarm $\{s^1(X^1, s_0^{1,i})\}_{i=1}^N$ is then carried to the $t=2$ stage.

For $t = 2, \ldots, T$, the algorithm generalizes as follows. First, combine $\{s_{t-1}^{t-1,i}\}_{i=1}^N$ with X_t in (11.13) to obtain the associated swarm $\{\upsilon_t(X^t, s_0^{t-1,i})\}_{i=1}^N$. Note that $s_{t-1}^{t-1,i}$ is the most recently obtained component of the sequence $s^{t-1}(X^{t-1}, s_0^{t-1,i})$. Using this swarm, approximate the integral for period t as

$$\int p(\upsilon_t(X^t, s_0))p(s_0|X^t)ds_0 \approx \frac{1}{N}\sum_{i=1}^{N} p(\upsilon_t(X^t, s_0^{t-1,i})). \tag{11.28}$$

Next, compute the weight assigned to $s_{t-1}^{t-1,i}$ as

$$q_t^i = \frac{p(\upsilon_t(X^t, s_0^{t-1,i}))}{\sum_{j=1}^N p(\upsilon_t(X^t, s_0^{t-1,j}))}. \tag{11.29}$$

Then draw from $\{s^{t-1}(X^{t-1}, s_0^{t-1,i})\}_{i=1}^N$ with replacement, using q_t^i as the probability assigned to the attainment of $s^{t-1}(X^{t-1}, s_0^{t-1,i})$. Denote the resulting swarm as $\{s^{t-1}(X^{t-1}, s_0^{t,i})\}_{i=1}^N$; this represents a set of drawings obtained from $p(s_0|X^t)$. Denote the most recently obtained component of $s^{t-1}(X^{t-1}, s_0^{t,i})$ as $s_{t-1}^{t,i}$. Combine this second swarm with X_t in (11.13) to obtain the associated swarm $\{\upsilon_t(X^t, s_0^{t,i})\}_{i=1}^N$. For each particle $s_{t-1}^{t,i}$ and its associated shock $\upsilon_t(X^t, s_0^{t,i})$, obtain the corresponding value of $s_t^{t,i}$ using the updating equation (11.9). Augment $s^{t-1}(X^{t-1}, s_0^{t,i})$ with $s_t^{t,i}$ to establish

$$s^t(X^t, s_0^{t,i}) = \left[s_t^{t,i}\ s_{t-1}^{t,i} \ldots s_1^{t,i}\ s_0^{t,i}\right]'.$$

Repeat these steps until $t = T$.

Having completed all T steps, the approximation to the integral we seek is given by

$$L(X^T|\mu) \approx \prod_{t=1}^{T} \left[\frac{1}{N} \sum_{i=1}^{N} p(\upsilon_t(X^t, s_0^{t-1,i})) \right]. \tag{11.30}$$

In turn, the $(T+1) \times 1$ sequence

$$\overline{s^T(X^T, s_0^T)}_N = \frac{1}{N} \sum_{i=1}^{N} s^T(X^T, s_0^{T,i}) \tag{11.31}$$

represents "smoothed" values of the state variable implied by X^T, conditional on the model. Finally,

$$\overline{\upsilon^T(X^T, s_0^T)}_N = \frac{1}{N} \sum_{i=1}^{N} \upsilon^T(X^T, s_0^{T,i}) \tag{11.32}$$

represents smoothed values of the structural shocks.

Exercise 11.2

Construct a procedure for evaluating $L(X^T|\mu)$ for the optimal growth model. Assume the TFP innovations ε_t are $iidN$ random variables with standard deviation σ_ε, which is a parameter to be estimated. For a given candidate μ, use a projection scheme to construct the policy function $c_t = c(z_t, k_t)$ in (11.18). As a suggestion, use the orthogonal collocation scheme implemented using `ara.prg` for this purpose. Organize the procedure as follows:

- Linearize the model, and use the elasticities σ_a and σ_k to construct starting values for $(\chi_{11}, \chi_{12}, \chi_{21}, \chi_{22})$ as described in chapter 10, section 10.2.6, equation (10.53).
- Construct the approximated policy function.
- Initiate the $t = 1$ step of the particle filter by obtaining 10,000 drawings of $s_0 = [z_0 \; k_0]'$ from a uniform distribution ranging ± 10 standard deviations above and below the steady state values z^* and k^*.
- Approximate $L(X^T|\mu)$ using (11.26), (11.28), and (11.30).
- Obtain the smoothed series $\overline{a^T}_N$ and $\overline{k^T}_N$ using (11.31), and $\overline{\varepsilon^T}_N$ using (11.32).

11.2.3 Case 2: Measurement Error

Turning to the case involving the presence of measurement error, the model is given by

$$s_t = f(s_{t-1}, \upsilon_t) \tag{11.33}$$

$$X_t = g(s_t, u_t), \tag{11.34}$$

where the policy function $c(s_t)$ has once again been subsumed in (11.33) and/or (11.34). The distributions of υ_t and u_t are given by $p(\upsilon_t)$ and $p(u_t)$; also, υ_t and u_t are taken as *iid*, and independent from each other. As usual, the parameterizations of $f(s_{t-1}, \upsilon_t)$, $g(s_t)$, $p(\upsilon_t)$, and $p(u_t)$ are determined by the specification of μ.

As an example, consider the following representation of the optimal growth model. The transition equations are as given above:

$$\ln(a_t) = \rho \ln(a_{t-1}) + \varepsilon_t, \tag{11.35}$$

$$k_t = a_{t-1} k_{t-1}^{\alpha} - c(a_{t-1}, k_{t-1}) + (1 - \delta) k_{t-1}. \tag{11.36}$$

But in this case, suppose output and investment are available as observable variables, each of which is measured with error. Then the observation equations are given by

$$y_t = a_t k_t^{\alpha} + u_{yt} \tag{11.37}$$

$$i_t = a_t k_t^{\alpha} - c(a_t, k_t) + u_{it}. \tag{11.38}$$

LIKELIHOOD CONSTRUCTION

To construct the likelihood function $L(X^T|\mu)$ in this case, (11.33) and (11.34) are used to obtain an expression for u_t as a function of X^t, υ^t, s_0, and μ. The probability assigned to this behavior by $p(u_t)$ provides the basis upon which the probability associated with the corresponding specification of μ is assessed. Conditionality on υ^t and s_0 is eliminated by integrating over the distribution $p(\upsilon^t, s_0|X^t)$; integration is once again facilitated via use of a particle filter.

To begin, solve for u_t using (11.33) and (11.34) to obtain an expression of the form

$$u_t = u(X_t, \upsilon_t, s_{t-1}). \tag{11.39}$$

Coupled with the transition equation (11.33), a recursive scheme for constructing $u_t(X^t, \upsilon^t, s_0)$ and $s_t(X^t, \upsilon^t, s_0)$, $t = 1, \ldots, T$, is as follows. For $t = 1$, combine a given s_0 and υ_1 with X_1 in (11.39) to obtain the implied u_1. Then combine s_0 and υ_1 in (11.33) to obtain s_1. For $t = 2, \ldots, T$, the value of s_{t-1} obtained in the previous step is combined with υ_t and

X_t in (11.39) to obtain the implied u_t, and with υ_t in (11.33) to obtain the implied s_t. The result is the pair of sequences $u^T(X^T, \upsilon^T, s_0) \equiv \{u_t(X^t, \upsilon^t, s_0)\}_{t=1}^T$ and $s^T(X^T, \upsilon^T, s_0) \equiv \{s_t(X^t, \upsilon^t, s_0)\}_{t=0}^T$.

Because u_t is taken as *iid*, the likelihood function for X^T, which at this point is conditional upon υ^T and s_0, is given by the product of individual likelihoods

$$L(X^T|\upsilon^T, s_0, \mu) = \prod_{t=1}^{T} p(u_t(X^t, \upsilon^t, s_0)). \tag{11.40}$$

To eliminate conditionality upon υ^T and s_0, $L(X^T|\upsilon^T, s_0, \mu)$ is integrated over the sequence of conditional distributions $p(\upsilon^t, s_0|X^t)$, $t = 1, \ldots, T$:

$$L(X^T|\mu) = \prod_{t=1}^{T} \iint p(u_t(X^t, \upsilon^t, s_0)) p(\upsilon^t, s_0|X^t) d\upsilon_t ds_0. \tag{11.41}$$

Exercise 11.3

Derive the expression $u_t = u(X_t, \upsilon_t, s_{t-1})$ in (11.39) for the optimal growth model, and use this expression to sketch the construction of $u^T(X^T, \varepsilon^T, s_0)$, $a^T(X^T, \varepsilon^T, s_0)$, and $k^T(X^T, \varepsilon^T, s_0)$.

LIKELIHOOD EVALUATION VIA THE PARTICLE FILTER

As with the case of no measurement error, inferences made conditionally upon s_0 (and additionally, υ^T) do not require the calculation of an integral in the likelihood evaluation step. All that is required is the construction of $u^T(X^T, \upsilon^T, s_0)$, which immediately enables the evaluation of $L(X^T|\upsilon^T, s_0, \mu)$ in (11.40). However, if s_0 and υ^T are unknown, we must approximate the sequence of integrals

$$\iint p(u_t(X^t, \upsilon^t, s_0)) p(\upsilon^t, s_0|X^t) d\upsilon_t ds_0,$$

bringing the particle filter into play.

Regarding notation, let $s_0^{t,i}$ continue to denote the i^{th} of N drawings of s_0 obtained from the conditional distribution $p(\upsilon^t, s_0|X^t)$, and $s_0^{0,i}$ the i^{th} of N drawings obtained from the unconditional distribution $p(s_0)$. In addition, let $\upsilon_t^{l,i}$ denote the i^{th} of N drawings of υ_t obtained from the conditional distribution $p(\upsilon^t, s_0|X^l)$, and $\upsilon_t^{0,i}$ from the unconditional distribution $p(\upsilon_t)$. For $t = 1, \ldots, T$, we will approximate the required integrals numerically by obtaining conditional drawings of $s_0^{t,i}$ and $\upsilon_t^{t,i}$ from $p(\upsilon^t, s_0|X^t)$. Once again, it is not possible to obtain these drawings directly from

$p(\upsilon^t, s_0|X^t)$; instead, the particle filter will be implemented to circumvent this problem.

The filter for $t = 1$ is initiated by obtaining a swarm of drawings $\{s_0^{0,i}\}_{i=1}^N$ from the unconditional distribution $p(s_0)$. This is augmented with an additional swarm of drawings $\{\upsilon_1^{0,i}\}_{i=1}^N$ obtained from the unconditional distribution $p(\upsilon_t)$. (Recall that the parameterization of $p(\upsilon_t)$ is determined by the specification of μ.) Combining particle pairs $(s_0^{0,i}, \upsilon_1^{0,i})$ with $X_1 \equiv X^1$ in (11.39) yields the associated sequence $\{u_1(X^1, \upsilon_1^{0,i}, s_0^{0,i})\}_{i=1}^N$. Note that the likelihood value associated with $(s_0^{0,i}, \upsilon_1^{0,i})$ is given by $p(u_1(X^1, \upsilon_1^{0,i}, s_0^{0,i}))$. Therefore, the integral for $t = 1$ can be approximated using

$$\iint p(u_1(X^1, \upsilon^1, s_0))p(\upsilon^1, s_0|X^1)d\upsilon_1 ds_0 \approx \frac{1}{N}\sum_{i=1}^{N} p(u_1(X^1, \upsilon_1^{0,i}, s_0^{0,i})).$$

Next, let q_1^i denote the relative likelihood value associated with $(s_0^{0,i}, \upsilon_1^{0,i})$, where relativity is with respect to its counterparts in the swarm:

$$q_1^i = \frac{p(u_1(X^1, \upsilon_1^{0,i}, s_0^{0,i}))}{\sum_{j=1}^N p(u_1(X^1, \upsilon_1^{0,j}, s_0^{0,j}))}. \tag{11.42}$$

Also, let $\{s_0^{1,i}, \upsilon_1^{1,i}\}_{i=1}^N$ denote a second sequence of drawings obtained by drawing with replacement from the original swarm, with the probability of obtaining the i^{th} particle pair determined by q_1^i. As in the case of no measurement error, this second swarm of particles once again represents a drawing from the conditional distribution $p(\upsilon^1, s_0|X^1)$.

To advance to the $t = 2$ stage, use each particle pair $(s_0^{1,i}, \upsilon_1^{1,i})$ to obtain the corresponding value of $s_1^{1,i}$ using the updating equation (11.33). Combine this with $s_0^{1,i}$ to establish the sequence

$$s^1(X^1, \upsilon_1^{1,i}, s_0^{1,i}) = \left[s_1^{1,i}\ s_0^{1,i}\right]'.$$

Also, use $\upsilon_1^{1,i}$ to establish the sequence

$$\upsilon^1(X^1, \upsilon_1^{1,i}, s_0^{1,i}) = \left[\upsilon_1^{1,i}\right].$$

These sequences will ultimately contain $T + 1$ and T components, respectively, representing the time series of state variables and structural shocks implied by X^T, conditional on the model. The swarm of sequences

$$\left\{s^1(X^1, \upsilon_1^{1,i}, s_0^{1,i}),\ \upsilon^1(X^1, \upsilon_1^{1,i}, s_0^{1,i})\right\}_{i=1}^N$$

is then carried to the $t = 2$ stage.

For $t = 2, \ldots, T$, the algorithm generalizes as follows. First, combine $\{s_{t-1}^{t-1,i}\}_{i=1}^{N}$ with X_t and an additional swarm of drawings $\{\upsilon_t^{t-1,i}\}_{i=1}^{N}$ obtained from the unconditional distribution $p(\upsilon_t)$ in (11.39) to obtain the associated swarm

$$\left\{u_t(X^t, \upsilon_t^{t-1,i}, s_0^{t-1,i})\right\}_{i=1}^{N}.$$

Note that $s_{t-1}^{t-1,i}$ is the most recently obtained component of the sequence $s^{t-1}(X^{t-1}, \upsilon_{t-1}^{t-1,i}, s_0^{t-1,i})$. Augment the sequence $\upsilon^{t-1}(X^{t-1}, \upsilon_{t-1}^{t-1,i}, s_0^{t-1,i})$ with $\upsilon_t^{t-1,i}$ to construct the new sequence

$$\upsilon^t(X^{t-1}, \upsilon_t^{t-1,i}, s_0^{t-1,i}) = \left[\upsilon_t^{t-1,i}\ \upsilon_{t-1}^{t-1,i} \ldots \upsilon_1^{t-1,i}\right]'.$$

The likelihood value associated with $(s_0^{t-1,i}, \upsilon_t^{t-1,i})$ is given by $p(u_t(X^t, \upsilon_t^{t-1,i}, s_0^{t-1,i}))$, thus the integral for period t is approximated as

$$\iint p(u_t(X^t, \upsilon^t, s_0))p(\upsilon^t, s_0|X^t)d\upsilon_t ds_0 \approx \frac{1}{N}\sum_{i=1}^{N} p(u_t(X^t, \upsilon_t^{t-1,i}, s_0^{t-1,i})).$$

Next, compute the weight assigned to $(s_{t-1}^{t-1,i}, \upsilon_t^{t-1,i})$ as

$$q_t^i = \frac{p(u_t(X^t, \upsilon_t^{t-1,i}, s_0^{t-1,i}))}{\sum_{j=1}^{N} p(u_t(X^t, \upsilon_t^{t-1,j} s_0^{t-1,j}))}. \tag{11.43}$$

Then draw from

$$\left\{s^{t-1}(X^{t-1}, \upsilon_{t-1}^{t-1,i}, s_0^{t-1,i}),\ \upsilon^t(X^{t-1}, \upsilon_t^{t-1,i}, s_0^{t-1,i})\right\}_{i=1}^{N}$$

with replacement, using q_t^i as the probability assigned to the attainment of the i^{th} particle pair. Denote the resulting swarm of sequences as

$$\left\{s^{t-1}(X^{t-1}, \upsilon_{t-1}^{t,i}, s_0^{t,i}),\ \upsilon^t(X^{t-1}, \upsilon_t^{t,i}, s_0^{t,i})\right\}_{i=1}^{N};$$

this represents a set of drawings obtained from the conditional distribution $p(\upsilon^t, s_0|X^t)$. Finally, for each particle pair $(s_{t-1}^{t,i}, \upsilon_t^{t,i})$ and its associated error $u_t(X^t, \upsilon_t^{t,i}, s_0^{t,i})$, obtain the corresponding value of $s_t^{t,i}$ using the updating equation (11.33). Augment $s^{t-1}(X^{t-1}, \upsilon_{t-1}^{t,i}, s_0^{t,i})$ with $s_t^{t,i}$ to establish

$$s^t(X^t, \upsilon_t^{t,i}, s_0^{t,i}) = \left[s_t^{t,i}\ s_{t-1}^{t,i} \ldots s_1^{t,i}\ s_0^{t,i}\right]'.$$

The swarm of sequences

$$\left\{s^t(X^t, \upsilon_t^{t,i}, s_0^{t,i}),\ \upsilon^t(X^t, \upsilon_t^{t,i}, s_0^{t,i})\right\}_{i=1}^N$$

is then carried to the next stage. Repeat these steps until $t = T$.

Having completed all T steps, the approximation to the integral we seek is given by

$$L(X^T|\mu) \approx \prod_{t=1}^{T}\left[\frac{1}{N}\sum_{i=1}^{N} p(u_t(X^t, \upsilon_t^{t-1,i}, s_0^{t-1,i}))\right]. \tag{11.44}$$

In turn, the $(T+1)\times 1$ sequence

$$\overline{s^T(X^T, \upsilon^T, s_0^T)}_N = \frac{1}{N}\sum_{i=1}^{N} s^T(X^T, \upsilon^{T,i}, s_0^{T,i}) \tag{11.45}$$

represents "smoothed" values of the state variable implied by X^T, conditional on the model. Also,

$$\overline{\upsilon^T(X^T, \upsilon^T, s_0^T)}_N = \frac{1}{N}\sum_{i=1}^{N} \upsilon^T(X^T, \upsilon^{T,i}, s_0^{T,i}) \tag{11.46}$$

represents smoothed values of the structural shocks. Finally,

$$\overline{u^T(X^T, \upsilon^T, s_0^T)}_N = \frac{1}{N}\sum_{i=1}^{N} u^T(X^T, \upsilon^{T,i}, s_0^{T,i}) \tag{11.47}$$

represents smoothed values of the measurement errors.

Exercise 11.4

Repeat exercise 11.3 for the version of the optimal growth model featuring measurement error.

11.2.4 Approximating the Unconditional Distribution of s_0

As we have described, the particle filter is initiated by obtaining a swarm of drawings $\{s_0^{0,i}\}_{i=1}^N$ from the unconditional distribution $p(s_0)$. Here we briefly describe how an approximation of this distribution can be obtained using the log-linear approximation of the underlying model.

Recall that log-linear model approximations are of the form

$$\begin{aligned} x_t &= Fx_{t-1} + G\upsilon_t \\ &= Fx_{t-1} + e_t, \end{aligned} \tag{11.48}$$

where the model variables x_t are represented as logged deviations from steady state values (e.g., $\widetilde{a}_t = \ln\left(\frac{a_t}{a^*}\right)$, and thus $a_t = a^* e^{\widetilde{a}_t}$). With $Ee_t e_t' = Q$, the variance-covariance matrix of x_t, denoted as $\Gamma(0)$, is given by

$$vec[\Gamma(0)] = [I - F \otimes F]^{-1} \, vec[Q],$$

where $\otimes$ denotes the Kronecker product (see chapter 4 for details).

Unconditionally then, according to the log-linearized model, the distribution of the state variables included in x_t have an expected value of 0 and variance-covariance matrix $\Gamma_s(0)$, which denotes the sub-matrix of $\Gamma(0)$ that corresponds with the state variables of the model. For example, in the case of the optimal growth model $\Gamma_s(0)$ is the 2×2 submatrix of $\Gamma(0)$ corresponding with $(\widetilde{a}_t, \widetilde{k}_t)$. Moreover, $(\widetilde{a}_t, \widetilde{k}_t)$ will be Normally distributed, following the distributional assumption made for TFP innovations. In sum,

$$p\left(\widetilde{a}_t, \widetilde{k}_t\right) \sim N(0, \Gamma_s(0)).$$

Drawings obtained from this distribution can be transformed into drawings of (a_t, k_t) via $a_t = a^* e^{\widetilde{a}_t}$ and $k_t = k^* e^{\widetilde{k}_t}$.

Exercise 11.5

Repeat exercises 11.3 and 11.4 by replacing the uniform distribution specified for $p(s_0)$ with the specification obtained using the log-linear approximation.

11.2.5 Data Alignment

As is the case in working with log-linear model approximations, it is important to align the actual data with their theoretical counterparts in working with nonlinear model approximations. We conclude this subsection with a brief discussion of this issue.

Recall that in working with log-linear approximations, the theoretical data are measured in terms of logged deviations from steady state values. Therefore, interpreting trend trajectories observed in the actual data as representing steady state behavior, the actual data are typically logged and detrended prior to empirical analysis.

In contrast, in working with nonlinear model approximations, it is typically the case that the theoretical variables are represented in terms of levels of variables that exhibit stationary fluctuations around steady state values. In such cases, the actual data should be transformed accordingly. For example, return to the specific case of the optimal growth model, which is to be estimated using output, consumption, and investment as observable variables. This model carries two implications regarding trend behavior. First, the variables follow a common trend. Second, long-run ratios of the variables (i.e., the relative heights of trend lines) should align with steady state ratios predicted by the model. Therefore, by detrending the data while preserving their relative means, the data can not only be aligned with the model, but their relative means can be used to help identify the model's structural parameters.

One means of implementing this approach to alignment is as follows. Begin by eliminating the trend components of logged values of each series. This may be accomplished, for example, by applying the Hodrick-Prescott filter or by subtracting a linear trend estimated via OLS. (If it is desired that a common growth rate be imposed in this step, the GAUSS procedure `ct.prc` can be used for this purpose.) Let the logged, detrended variables be denoted as $\widehat{y}_t$, and so on. Next, construct new series $(\widetilde{y}_y, \widetilde{c}_t, \widetilde{i}_t)$ as

$$\widetilde{y}_t = e^{\widehat{y}_t}, \quad \widetilde{c}_t = \frac{\overline{c}}{\overline{y}} e^{\widehat{c}_t}, \quad \widetilde{i}_t = \frac{\overline{i}}{\overline{y}} e^{\widehat{i}_t}, \tag{11.49}$$

where $\overline{y}$ is the sample mean of y_t, and so on. The resulting series will be detrended, and ratios of their sample means will approximate their untransformed counterparts (and the level of $\widetilde{y}_t$, which serves as a numeraire, will approximately equal 1).

These transformed series are carried into the likelihood-evaluation stage of the analysis. However, for each candidate parameterization of the structural model μ, a final adjustment is necessary. This involves scaling the series so that the sample average of one of the variables matches the steady state value of its theoretical counterpart. For example, let $y_{ss}(\mu)$ represent the steady state value of y for a given parameterization μ. Then by scaling $\widetilde{y}_t$ by $y_{ss}(\mu)$, creating $\widetilde{\widetilde{y}}_t = y_{ss}(\mu)\widetilde{y}_t$, the sample average of the scaled series will approximate its steady state counterpart. The additional series $\widetilde{c}_t$ and $\widetilde{i}_t$ are also scaled by $y_{ss}(\mu)$, so that the relative sample averages of $(\widetilde{\widetilde{y}}_t, \widetilde{\widetilde{c}}_t, \widetilde{\widetilde{i}}_t)$ will continue to match their unadjusted counterparts. Then if the particular parameterization μ implies counterfactual ratios between y, c, and i, the sample means of $(\widetilde{\widetilde{c}}_t, \widetilde{\widetilde{i}}_t)$ will fail to align with their corresponding steady state values $c_{ss}(\mu)$ and $i_{ss}(\mu)$, and thus the value of the likelihood function associated with μ will reflect the failure of μ to account for this particular aspect of the data.

11.3 Linear Versus Nonlinear Model Approximation

As we have seen, linear and nonlinear model approximations serve as alternative points of departure for implementing structural models empirically. The choice between these two general alternatives involves a trade-off. One one hand, linear approximations are both easy to program and computationally inexpensive relative to their nonlinear counterparts. Moreover, the Kalman filter is both easy to program and computationally inexpensive relative to the particle filter. On the other hand, approximation errors associated with the use of nonlinear methods are in general less severe than those associated with the use of linear methods. In short, the trade-off involves a choice between simplicity and speed versus accuracy.

Methodological and computational advances have already served to help mitigate concerns regarding costs associated with the use of nonlinear model approximations. Undoubtedly, such concerns will continue to erode over time, although as we shall see below, costs at this point remain nontrivial. However, an outstanding question involves the relative benefits associated with their use. Specifically, the question is whether the superior accuracy associated with nonlinear model approximations is significant empirically: do nonlinear approximations yield substantive differences regarding the empirical question at hand? Unfortunately, the answer to this question is likely to be somewhat specific, both to the particular model under investigation and to the empirical question the model is being used to address. However, general guidance may emerge through experiments designed to address this question in specific contexts.

An example of such an experiment is provided by Fernandez-Villaverde and Rubio-Ramirez (2005), who conducted a likelihood-based analysis of an RBC model closely related to that introduced in chapter 5. In their experiment, they generated two artificial data sets using two alternative parameterizations of the model. For each data set, they obtained flat-prior posterior estimates of the model parameters using log-linear and nonlinear approximation methods, and compared the estimates they obtained to the actual parameters. They found that although point estimates associated with each approximation method were comparable, and provided close matches with their corresponding actual values, distributions obtained using the nonlinear model approximation they used were more tightly concentrated around these actual values.

To convey the flavor of their results, here we report on a similar exercise patterned after an extension of the Fernandez-Villaverde/Rubio-Rameriz experiment conducted by Fernandez-Villaverde, Rubio-Ramirez, and Santos (2006). The exercise focuses on maximum likelihood (ML) estimates obtained using an artificial data set generated using the optimal growth

model, parameterized so that an exact analytical solution of the policy function is available.

Specifically, the model specification is given by (11.35)–(11.38), with

$$(\alpha, \beta, \delta, \phi) = (0.33, 0.96, 1, 1),$$

so that the policy function is given by (10.38). The AR(1) representation for $\ln(a_t)$ was parameterized as

$$(\rho, \sigma_\varepsilon) = (0.8, 0.0067),$$

with the TFP innovations ε_t specified as being Normally distributed. Output and investment were treated as observable variables subject to measurement error. The errors associated with the observation of both variables were also specified as being Normally distributed, with standard deviations set to 0.5% of their corresponding steady state values. We worked with a sample size of 100, using data simulated as described in exercise 11.2.

ML estimates were obtained subject to the restriction that the steady state investment-output ratio corresponds with the sample mean of its empirical counterpart. With the steady state ratio given by

$$\frac{i^*}{y^*} = \alpha\beta,$$

this restriction was imposed using

$$\beta = \frac{1}{\alpha}\overline{\left(\frac{i}{y}\right)},$$

with $\overline{\left(\frac{i}{y}\right)}$ denoting the sample mean of $\left(\frac{i_t}{y_t}\right)$. This restriction was imposed primarily as a practical matter, because deviations from the restriction are assigned very little weight by the likelihood function under any representation of the model. (This illustrates the strong source of parameter identification that can arise from the preservation of relative means in the actual data.)

Our initial implementation of this exercise was programmed in GAUSS. However, here we confronted a computational hurdle: the evaluation of the likelihood function for a single candidate parameterization turned out to require approximately 24 seconds of CPU time on a 3 GHz Pentium 4 desktop computer using a 20,000-element particle swarm. In contrast, likelihood evaluation was accomplished at the rate of 15 per second in working with the log-linear approximation. This large computational cost prompted us to explore the payoff of switching from GAUSS to Fortran in executing this application. (This also follows Fernandez-Villaverde

TABLE 11.2
Maximum Likelihood Estimates

Parameter	*Actual Value*	*Log-Lin. Appx.*	*Nonlin. Appx.*	*Exact Policy Fcn.*
α	0.33	0.330 $(3.14e-05)$	0.327 $(1.11e-05)$	0.327 $(3.77e-05)$
β	0.96	0.962 $(9.16e-05)$	0.968 $(1.04e-04)$	0.968 $(3.52e-05)$
ρ	0.80	0.784 $(2.29e-02)$	0.794 $(5.10e-05)$	0.809 $(2.80e-05)$
σ_ε	0.0067	0.0063 $(5.21e-04)$	0.0061 $(1.01e-06)$	0.0061 $(1.08e-06)$
$\sigma_{u,y}$	0.00284	0.00539 $(4.96e-04)$	0.00309 $(2.77e-07)$	0.00316 $(6.90e-07)$
$\sigma_{u,i}$	0.0090	0.00484 $(5.12e-04)$	0.00086 $(7.81e-09)$	0.00087 $(6.76e-09)$
$\log L$:		782.75	917.74	917.78

Notes: Standard deviations are in parentheses.

and Rubio-Ramirez, 2005, who implemented their investigation using Fortran.)

Due to a considerable advantage in dealing with large particle swarms, the payoff turned out to be substantial: the cost of likelihood evaluation was reduced to approximately 4 seconds of CPU time using Fortran. (Roughly 87% of the CPU time required by GAUSS to achieve likelihood evaluation is devoted to implementation of the particle filter. In contrast, roughly 3 seconds is devoted to approximating the policy function.) In order to provide guidance for achieving the conversion from GAUSS to Fortran, code used to facilitate likelihood evaluation in this application is available in both programming languages at the textbook Web site.[1]

Table 11.2 presents three sets of parameter estimates. The first set was obtained by combining the log-linear model approximation with the Kalman filter; the second by combining a nonlinear model approximation with the particle filter; and the third by combining the exact policy function with the particle filter. The nonlinear model approximation was constructed using the orthogonal collocation scheme described in chapter 10, section 10.2 (for the case in which $\ln(a)$ follows (10.8)).

[1]GAUSS procedures used to accomplish likelihood evaluation are provided in `nonling.src`; corresponding Fortran procedures are provided in `nonlinf.src`. We are deeply indebted to Hariharan Dharmarajan for executing the transition to Fortran in this application.

Notice that the three sets of estimates obtained for $(\alpha, \beta, \rho, \sigma_\varepsilon)$ are quite similar, and correspond closely to their associated "true" values. However, the same is not true of the estimates obtained for $(\sigma_{u,y}, \sigma_{u,i})$ (denoting standard deviations of the measurement errors associated with the observation of y and i). Although the estimates obtained for $\sigma_{u,y}$ using the exact and nonlinear model approximations differ from their corresponding true value by approximately 10% (and by approximately 4% for $\sigma_{u,i}$), estimates obtained using the log-linear approximation differ roughly by factors of 2 for $\sigma_{u,y}$ and 5 for $\sigma_{u,y}$. In turn, whereas the difference in log-likelihood values observed in moving from the exact policy function to the nonlinear approximation is a mere 0.04, the difference in moving to the log-linear approximation is 135. The pattern of these results closely matches the findings of Fernandez-Villaverde and Rubio-Ramirez (2005) and Fernandez-Villaverde, Rubio-Ramirez, and Santos (2006).

Seeking to account for this Monte Carlo evidence, Fernandez-Villaverde, Rubio-Ramirez, and Santos (2006) have discovered that the source of likelihood differences arising from alternative approaches to model approximation can be traced to errors associated with policy function approximations. Specifically, they have shown that approximation errors in representing the policy function translate into errors in associated approximations of the likelihood function. Moreover, the mapping of errors from policy to likelihood functions is compounded by sample size: period by period, errors in approximating the policy function accumulate as the size of the sample expands. Thus second-order approximation errors in the policy function translate into first-order approximation errors in the likelihood function, and so on.

In working with nonlinear approximations of the policy function, this problem can be combated by working with approximations of the policy function that increase in quality with the sample size (e.g., by increasing the order of Chebyshev polynomials used in implementing orthogonal collocation approximations, or refining the grid over which the state space is divided in working with value-function and policy-function approximations). However, no such remedy is available in working with log-linear approximations: in this case, the quality of approximation is determined strictly by the proximity of the actual model representation to linearity.

It is too early to fully discern the implications this "compounding" problem carries for the future use of log-linear approximations as foundations for conducting likelihood-based empirical analyses. For example, suppose that in a given application the problem merely affected measurements of the absolute height of the likelihood function, but left measurements of its relative height at alternative candidate parameterizations intact. Then calculations of posterior means and ML point estimates would not be prone to the compounding problem, whereas cross-model likelihood comparisons

(e.g., posterior odds calculations or likelihood ratio tests) would. The mixed pattern of discrepancies between actual and estimated parameters that have been reported to date have left this question largely unresolved, and further research on this issue is clearly warranted.

But even if the compounding problem proves to be problematic in general, there remains an important role for log-linear approximations in conducting likelihood analyses. This is the case along at least two dimensions. First, recall from chapter 10, section 10.2, the important role played by the log-linear approximation of the optimal growth model in providing starting values for implementing the orthogonal collocation approximation of the policy function. In empirical applications that involve repeated approximations of the policy function for each candidate parameterization of the model, the ability to obtain fast, accurate, and reliable approximations is critical. By providing an automated means of constructing effective starting values for this purpose, log-linear approximations serve as an instrumental input in the estimation process. Second, recall from section 11.2 that log-linear approximations can also be used to construct an approximation of the unconditional distribution $p(s_0)$ that is used to initiate the particle filter. This too is critical, because an inappropriate specification of this distribution can introduce an additional source of error in working with likelihood approximations obtained using the particle filter.

In sum, recent research suggests that the relatively large approximation errors associated with log-linear model representations may limit the usefulness of these representations as foundations upon which full-information empirical analyses involving DSGE models are conducted. However, more research on this subject is needed before general conclusions can be drawn. But as we have illustrated, log-linear approximations can serve to provide important inputs into the process of implementing nonlinear model representations empirically. Thus even as the state-of-the art advances towards a more widespread use of nonlinear model approximations, we envision a lasting complementary role for log-linear approximations in conducting structurally based empirical research.